



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/585,101

06/30/2006

Eric Vetillard

VETILLARD2

5750

1444 7590 06/09/2009
BROWDY AND NEIMARK, P.L.L.C.
624 NINTH STREET, NW
SUITE 300
WASHINGTON, DC 20001-5303

EXAMINER

TECKLU, ISAAC TUKU

ART UNIT

PAPER NUMBER

2192

MAIL DATE

DELIVERY MODE

06/09/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/585,101	Applicant(s) VETILLARD ET AL.	
	Examiner ISAAC T. TECKLU	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 March 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-18 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 1-18 have been examined.

Response to Arguments

2. Applicant's arguments with respect to claims 1-18 have been considered but are moot in view of the new ground(s) of rejection. See Liang (US 7,120,572 B1), art made of record.

Drawings

3. The subject matter of this application admits of illustration by a drawing to facilitate understanding of the invention. Applicant is required to furnish a drawing under 37 CFR 1.81(c). No new matter may be introduced in the required drawing. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d).

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claim 16 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 16 is directed to a system. However, as recited, the system is reasonably interpreted as entirely software, which amounts to descriptive material per se. The system is not supported by hardware such as tangible computer storage or execution engine, which would enable one skill in the

Art Unit: 2192

art to construe that the system, is built from tangible product to carry out any functionality being conveyed from the claim. Thus, the system is software *per se* and therefore is not being tangibly embodied in a manner as to be executable. See MPEP § 2106.01.

Claims 17-18 are rejected for failing to cure the deficiencies of the above rejected non-statutory claim 16 above.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

6. Claims 1-18 are rejected under 35 U.S.C. 102(e) as being anticipated by Liang (US 7,120,572 B1).

Per claim 1 (Currently Amended), Liang teaches a method for determining the operational characteristics of a program, comprising a verification procedure comprising the following steps (see at least e.g. FIG. 5, steps 300-306 “At authoring system Pre- verify class file...” and related text):

a first step comprising:

expressing the operational characteristics of the program as functions dealing with occurrences or sequences of occurrences of events occurring which may occur during possible executions of the program (see at least col.5:1-15 “... full program verifier for verifying whether or

not a specified program satisfies certain predefined integrity criteria...”), said events being able to deal with particular operations, particular values of data (see at least e.g. FIG. 5, 308-306 and related text), at particular program points and in particular states of the program (see at least col.10:30-45 “... verifier returns a verification failure value, ... to execute the specified program... ”);

determining a possible level of precision with which these characteristics must be determined (see at least col.5:1-15 “... full program verifier for verifying whether or not a specified program satisfies certain predefined integrity criteria...”);

determining a possible set of particular contexts of execution in which the program will always be executed (see at least col.6:20-35 “... stack and local variable contents prior to their execution.... program verification process...” and col.7:1-20 “... program may contain errors involving the data type of operands not matching ... fail during execution...”);

determining possible operational specificities of a set of platforms on which the program will be executed (see at least col.6:35-55 “...programs executed on one will not be executed on the others... platform specific...”);

a second step of estimation, by program analysis, and in consideration of said possible level of precision, of said possible set of particular contexts of execution and of said possible operational specificities of platforms (see at least col.27:25-45 “... verify that execution of the program will not violate the data type restriction...”), at various points of the execution paths and under different execution conditions, of the states of the program and data handled by the program (see at least col.5:1-15 “... full program verifier for verifying whether or not a specified program satisfies certain predefined integrity criteria...”);

a third step for determining said operational characteristics, by means of the information extracted by said program analysis (see at least e.g. FIG. 7A-7B, 350-364, 380-392 and related text),

by the computation of said functions on the occurrences or particular sequences of occurrences of particular operations, dealing with particular values, at particular points of the program, (see at least col.14:55-65 "... detect any stack or register usage error during analysis... verification success flag..."), in particular states of the program, for the set of execution paths determined by analysis (see at least col.10:30-45 "... verifier returns a verification failure value, ... to execute the specified program...").

Per claim 2 (Currently Amended), Liang teaches wherein, in the case when the program is interactive and may depend on an undetermined number of dynamic values resulting from this interaction (see at least col.10:35-55 "... operand stack overflow and underflow checking...during program execution..."), the contexts of execution are given by a description abstracted from possible series of data representing said dynamic values (see at least e.g. FIG. 7A, 350 and related text).

Per claim 3 (Currently Amended), Liang teaches, wherein, in the case where the program is inserted into a framework of execution (see at least e.g. FIG. 7A, 356 and related text), said second step of estimation comprises static analysis which also take into account the semantics of this framework of execution, including the possible implicit interaction loops of the program (see at least col.26:15-45 "... get static field values...").

Per claim 4 (Currently Amended), Liang teaches wherein certain of said particular operations, which from events, accompanied by constraints on the values handled, the execution points (see at least col.6:20-35 "... stack and local variable contents prior to their execution....

program verification process..." and col.7:1-20 "... program may contain errors involving the data type of operands not matching ... fail during execution..."), and the statuses of the program are defined as one of the following actions (see at least col.10:35-55 "... operand stack overflow and underflow checking...during program execution..."): call to a given routine, access to a given variable, reading or writing on a given port, computation of a given arithmetic expression, completion of execution of the program or of a routine on a normal return or ending an exception (see at least col.26:15-45 "... get static field values...").

Per claim 5 (Currently Amended), Liang teaches wherein certain of said static analysis consist of abstract interpretations of the program (see at least col.26:15-45 "... get static field values..."), on abstract domains which may notably represent possible sets of values and symbolic expressions (see at least col.14:55-65 "... detect any stack or register usage error during analysis... verification success flag...").

Per claim 6, Liang teaches wherein said extracted information are represented by means of one or more of the following structures: status graph of the program, inheritance graph, graph of the routine calls of the program (see at least col.6:20-35 "... stack and local variable contents prior to their execution.... program verification process..." and col.7:1-20 "... program may contain errors involving the data type of operands not matching ... fail during execution..."), control flow chart of each routine of the program, structure of loops and catch-up of exceptions, structure of basic blocks, abstraction of the status of the program at an execution point (see at least col.26:15-35 "... call method... static method...").

Per claim 7, Liang teaches wherein said extraction of information does not apply to unnecessary information for determining the operational characteristics (see at least col.5:1-15 "... full program verifier for verifying whether or not a specified program satisfies certain predefined integrity criteria..."), both from the viewpoint of the amount of information extracted and from the precision of these pieces of information (see at least e.g. FIG. 7A, 350 and related text).

Per claim 8, Liang teaches wherein only major pieces of information among said extracted information are computed and saved and in that the other pieces of information are only computed when necessary for determining said operational characteristics (see at least col.6:20-35 "... stack and local variable contents prior to their execution.... program verification process..." and col.7:1-20 "... program may contain errors involving the data type of operands not matching ... fail during execution...").

Per claim 9 (Currently Amended), Liang teaches wherein the major pieces of information are information extracted at breakdown nodes of the code of routines in a graph of basic blocks (see at least col.5:50-60 instruction set is characterized by byte code instruction ... data type specific...) and in that the other pieces of information in the body of the basic blocks are recomputed by local analysis from information saved at the start and end of the corresponding block (see at least col.26:15-45 "... get static field values...").

Per claim 10 (Currently Amended), Liang teaches wherein said operational characteristics represent validity criteria and in that said determination establishes that the program is valid because it observes each of said criteria (see at least col.6:20-35 "... stack and local variable contents prior to

their execution.... program verification process..." and col.7:1-20 "... program may contain errors involving the data type of operands not matching ... fail during execution..."), or invalid because at least one of said criteria cannot be observed (see at least col.10:30-45 "... verifier returns a verification failure value, ... to execute the specified program... ").

Per claim 11, Liang teaches wherein said validity criteria express security or interoperability rules (see at least col.5:1-15 "... full program verifier for verifying whether or not a specified program satisfies certain predefined integrity criteria...").

Per claim 12, Liang teaches wherein said operational characteristics characterize resources which are consumed and functionalities which are exploited by the program during its execution (see at least col.7:1-20 "... program may contain errors involving the data type of operands not matching ... fail during execution...") and in that said determination provides an execution profile of the program (see at least col.15:55-65 "... actual execution of the program for each datum to be pushed...").

Per claim 13, Liang teaches wherein a computation of certain of said functions associated with the operational characteristics is performed during said static program analysis, as soon as certain of said pieces of information are extracted (see at least col.26:15-45 "... get static field values...").

Per claim 14 (Currently Amended), Liang teaches the method according to claim 10 for automatic filtering of a set of programs relative to a given set of validity criteria, wherein the

extraction of information by static program analysis is only completed once per program (see at least col.6:20-35 "... stack and local variable contents prior to their execution.... program verification process..." and col.7:1-20 "... program may contain errors involving the data type of operands not matching ... fail during execution...") and reused whenever necessary for determining whether the program observes said set of validity criteria (see at least col.10:30-45 "... verifier returns a verification failure value, ... to execute the specified program... ").

Per claim 15 (Currently Amended), Liang teaches method for distribution of applications ensuring that the applications observe validity criteria associated with the execution platforms of these applications, comprising filtering means designed such that, for any client desiring to accede to the applications for a certain execution platform (see at least col.5:1-15 "... full program verifier for verifying whether or not a specified program satisfies certain predefined integrity criteria..."), the applications are filtered by a verification procedure in accordance with the method according to any one of claims 1 to 12, only the applications which observe the validity criteria for said platform being presented to the client (see at least col.7:1-20 "... program may contain errors involving the data type of operands not matching ... fail during execution...").

Per claim 16 (Currently Amended), Liang teaches a system for multi- application execution ensuring that the applications observe given validity criteria, comprising:

an application analysis server, a server for validation of applications and a multi-application platform (see at least e.g. FIG. 1, 102, 105 and related text), and

means for ensuring, prior to loading or execution of an application on the platform (see at least e.g. FIG. 1, 102, 103, 105 and related text):

observance by this application of said validity criteria an extraction of information being carried out on the application analysis server and an evaluation of said validity criteria being carried out on the server for validation of applications (see at least col.5:1-15 "... full program verifier for verifying whether or not a specified program satisfies certain predefined integrity criteria..."), and

in the case when one of the validity criteria cannot be observed, a failure of loading or execution of the application a change of the status of the system and an emission of a sound or visual signal to alert of failure of loading or execution, the means for ensuring observance by said application of said validity criteria executing a procedure comprising the following steps (see at least e.g. FIG. 5, 308-306 and related text), at particular program points and in particular states of the program (see at least col.10:30-45 "... verifier returns a verification failure value, ... to execute the specified program... "):-

first step comprising:

expressing the validity criteria of the program as functions dealing with occurrences or sequences of occurrences of events occurring executions of the program, said events being able to deal with particular operations, particular values, at particular program points and in particular states of the program (see at least e.g. FIG. 5, 308-306 and related text), at particular program points and in particular states of the program (see at least col.10:30-45 "... verifier returns a verification failure value, ... to execute the specified program... ");

determining a possible level of precision with which these validity criteria must be determined (see at least col.5:1-15 "... full program verifier for verifying whether or not a specified program satisfies certain predefined integrity criteria...");

determining a possible set of particular contexts of execution in which the program will always be executed (see at least col.6:20-35 "... stack and local variable contents prior to their

execution.... program verification process..." and col.7:1-20 "... program may contain errors involving the data type of operands not matching ... fail during execution...");

determining possible operational specificities of a set of platforms on which the program will be executed (see at least col.6:35-55 "...programs executed on one will not be executed on the others... platform specific...");

a second step of estimation, by program analysis, and in consideration of said possible level of precision, of said possible set of particular contexts of execution and of said possible operational specificities of platforms (see at least col.27:25-45 "... verify that execution of the program will not violate the data type restriction..."), of information relating to the structure of the program, the possible execution paths of the program and to the values of possible data, at various points of the execution paths and under different execution conditions, of the states and data handled by the program (see at least col.5:1-15 "... full program verifier for verifying whether or not a specified program satisfies certain predefined integrity criteria...");

a third step for determining said validity criteria, by means of the information extracted by said program analysis (see at least e.g. FIG. 7A-7B, 350-364, 380-392 and related text), by the computation of said functions on the occurrences or particular sequences of occurrences of particular operations (see at least col.14:55-65 "... detect any stack or register usage error during analysis... verification success flag..."), dealing with particular values, at particular points of the program, in particular states of the program, for the set of execution paths determined by analysis (see at least col.10:30-45 "... verifier returns a verification failure value, ... to execute the specified program... ").

Per claim 17, Liang teaches wherein the server for validation of applications is executed on the multi- application platform, the application analysis server executing outside the platform (see at least col.6:35-55 "...programs executed on one will not be executed on the others... platform specific...").

Per claim 18, Liang teaches wherein the application analysis server and the server for validation of applications are executed on the multi-application platform (see at least e.g. FIG. 1, 102, 105 and related text).

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ISAAC T. TECKLU whose telephone number is (571) 272-7957. The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Isaac T Tecklu/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192